Desconocido

### ¿Por qué utilizar clases y objetos?

Ahora hemos añadido funciones a nuestras clases, pero ¿por qué utilizar clases y objetos cuando podrías escribir funciones normales llamadas breathe , move , eat\_food , etc.?

Para responder a esa pregunta, utilizaremos nuestra jirafa llamada Reginald, que creamos antes como un objeto de la clase Giraffe , de la siguiente manera:

>>> reginald = Giraffe()

Como reginald es un objeto, podemos llamar (o ejecutar) funciones proporcionadas por la clase Giraffe y sus clases padre. Llamamos a funciones de un objeto utilizando el operador punto ( . ) y el nombre de la función. Para decirle a Reginald que se mueva o que coma, podemos llamar a las funciones de la siguiente manera:

>>> reginald = Giraffe()  
>>> reginald.move()  
>>> reginald.eat\_leaves\_from\_trees()

Supongamos que Reginald tiene una amiga jirafa llamada Harriet. Vamos a crear otro objeto Giraffe llamado harriet :

>>> harriet = Giraffe()

Como estamos utilizando objetos y clases, podemos decirle a Python de qué jirafa estamos hablando cuando queramos ejecutar la función move . Por ejemplo, si queremos hacer que Harriet se mueva pero dejar a Reginald en su sitio, podríamos llamar a la función move utilizando nuestro objeto harriet , así:

>>> harriet.move()

En este caso, sólo se movería Harriet.

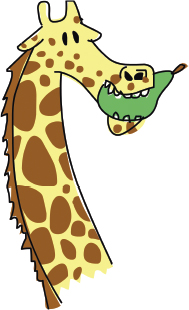
Cambiemos un poco nuestras clases para que esto sea más evidente. Añade una sentencia print a cada función en lugar de utilizar pass :

>>> class Animal(Animate):  
 def breathe(self):  
 print('breathing')  
 def move(self):  
 print('moving')  
 def eat\_food(self):  
 print('eating food')  
>>> class Mammal(Animal):  
 def feed\_young\_with\_milk(self):  
 print('feeding young')  
  
>>> class Giraffe(Mammal):  
 def eat\_leaves\_from\_trees(self):  
 print('eating leaves')

Ahora, cuando creamos nuestros objetos reginald y harriet y llamamos a funciones sobre ellos, podemos ver que realmente ocurre algo:

>>> reginald = Giraffe()  
>>> harriet = Giraffe()  
>>> reginald.move()  
moving  
>>> harriet.eat\_leaves\_from\_trees()  
eating leaves

En las dos primeras líneas, creamos las variables reginald y harriet , que son objetos de la clase Giraffe . A continuación, llamamos a la función move sobre reginald , y Python imprime moving en la línea siguiente. Del mismo modo, llamamos a la función eat\_leaves\_from\_trees sobre harriet , y Python imprime eating leaves . Si se tratara de jirafas reales, en lugar de objetos de un ordenador, una jirafa estaría caminando y la otra comiendo.



**NOTA**

*Las funciones definidas para las clases se llaman en realidad methods . Los términos son casi intercambiables, salvo que los métodos sólo pueden invocarse sobre objetos de una clase. Otra forma de decir esto es que un método está asociado a una clase, pero una función no. Dado que son casi lo mismo, en este libro utilizaremos el término function* .

[anterior](ch08_6.html)[Subtema 7 de 14: (Ver todo)](ch08.html)[siguiente](ch08_8.html)